AR-001-227

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

ELECTRONICS RESEARCH LABORATORY

TECHNICAL REPORT

ERL-0021-TR

A SELECTIVE DATASET BACKUP SCHEME FOR THE DRCS CENTRAL COMPUTER

J.C. Gwatking

S U M M A R Y

This report describes the approach taken by the DRCS Computing Centre to maintaining adequate backup facilities while the quantity of data stored on-line continues to increase. The aim is to provide the required level of service to users with minimum operational overhead. This has been accomplished by the development of a software system to control the backup process. The basis of this scheme is the ability to determine which datasets have been altered and to select only these for backup.

Approved for Public Release

POSTAL ADDRESS: Chief Superintendent, Electronics Research Laboratory, Box 2151, G.P.O., Adelaide, South Australia, 5001.

TABLE OF CONTENTS

LIST OF APPENDICES

1. INTRODUCTION

The DRCS Computing Centre has always placed great importance on the protection of user data against accidental or malicious damage. It has assumed the responsibility for maintaining adequate backups of datasets, rather than burdening the user with this task - although the DRCS data migration scheme includes a user-driven backup capability(ref.1).

Up to September 1977 all on-line user data was stored on seven IBM 3330-1 disk packs. Until the development of the scheme described in this report the IBM utility program IEHDASDR(ref.7) was used for taking backups. Because of the requirement to provide good recovery capabilities for all datasets the seven 3330's were dumped onto magnetic tape each night. The backups were kept for the previous three days and the previous two Thursdays. This meant that the Centre could recover up to five versions of a dataset, although they might all be the same. The earliest was at most two weeks old and this soon proved inadequate to meet all requests for data recovery.

IBM 3330-1 disk drives were also used to store Operating System datasets. They were, and continue to be, dumped to magnetic tape once a week, on Thursday night.

This report highlights the problems associated with the existing backup procedures, using IEHDASDR, and the main features of the scheme designed to replace them, as it existed at the end of April 1978. The report assumes some understanding of the terminology associated with IBM computer systems. A glossary is included for clarification of the more important terms and mnemonics used.

2. THE DEFICIENCIES OF THE FORMER APPROACH TO BACKUPS

The main problem with the IEHDASDR backups is the difficulty in restoring a dataset. First the backup tape containing the dataset must be located. Then the entire contents must be restored to a spare disk volume. Finally the dataset can be copied to the required user volume. Unfortunately it takes a considerable amount of time to organize the use of a spare disk drive and to prepare and execute the necessary batch jobs.

Often the user requesting the service is unsure as to which backup version he requires, so that the whole process may need to be repeated, perhaps several times.

This raises a second problem. Because of the limited number of backups kept, the version of the dataset the user requires may not be available. This may be the case, for example, when the request is prompted by the contents of a dataset access report. Such reports are issued to users fortnightly and they list accesses to the users' datasets by other, unauthorized users. By the time the report is received and studied the dataset may have been damaged for several weeks, so that some backup copy at least this old is required.

The third deficiency of the IEHDASDR backups is the operational overhead they impose. The Computing Centre began replacing the IBM 3330-1 drives by IBM 3350 drives from about the beginning of September 1977. The latter can store up to three times the data of a 3330-1 and therefore the operational overhead of taking full backups of each user disk every night is potentially trebled. Since the elapsed time to dump the seven 3330's is between 25 and 30 minutes the time for the same number of 3350's could be over an hour, an unacceptable cost for an infrequently used facility. In addition it is unlikely that there will be a spare 3350 drive upon which to restore a full backup.

These shortcomings to the use of IEHDASDR for backing up data prompted a study to determine the requirements of a backup scheme at DRCS and how these could be met.

3. THE REQUIREMENTS OF A NEW BACKUP SCHEME

The study resulted in the following guidelines for a new backup scheme.

(i) The operational overhead in taking backups should be considerably less than what would be required by using IEHDASDR. An elapsed time of less than 30 minutes would be acceptable.

(ii) The new scheme should provide better service to users than the old one, in terms of the availability of backup copies. In particular, the following guidelines were proposed as a minimum guaranteed level of service.

(a) The scheme should retain all backups of all datasets for at least a fortnight. That is, it should be possible to restore any dataset to its state at backup time on any night during the previous fortnight.

(b) In addition several older copies of all datasets should be available. Copies at fortnightly intervals, with the earliest at least six weeks old, would be sufficient.

With this degree of availability the scheme should be able to honour even the most demanding requests for recovery encountered in the past.

(iii) The process for restoring a dataset should be much simpler and faster than using the IEHDASDR backups.

(iv) The number of magnetic tapes used to meet the requirements of (ii) should not be excessive.

(v) The requirement to occasionally restore a complete disk volume should not be overlooked. This of course is the advantage of the IEHDASDR backups. Complete volumes can quickly be restored both on-line and stand-alone. While we can tolerate less efficiency in this area because of the infrequency of use (a full backup of a user 3330 volume has been required only twice in two years), the capability must be provided.

(vi) The system should be as flexible as possible, so that parametric as well as logic changes can be easily implemented.

(vii) There should be little or no operator or programmer involvement needed to run the system.

Under these constraints a proposal for a set of software to manage a new backup scheme was prepared and approved in November 1976. Program specifications were frozen and coding commenced in February 1977. Testing was completed by the end of June 1977 and parallel running on the 3330 disks began. The old system had been gradually phased out by September, at about the time the 3350's started to arrive.

4. OVERVIEW OF THE NEW SCHEME

(i) The new scheme uses a technique of selective backups to meet the design objectives. Full backups of each volume are taken only periodically (say once a fortnight). On the intervening nights only datasets that have been altered or created during that day are backed up. The current contents of an unaltered dataset will still be reflected by the most

recent copy of it on the backup tapes. Since an average of only 14% of datasets on the 3330 volumes at DRCS change each day this technique has obvious advantages over any scheme that uses full backups exclusively.

(ii) SMF data(ref.8) is used to determine which datasets have been updated, and therefore forms the basis of the scheme. SMF is a component of the Operating System that monitors and records system events, including dataset accesses. This information is extensively used at DRCS and is extremely important to the operation of the computer system.

(iii) The new software maintains a catalogue identifying the tape volumes containing the individual copies of each dataset, and when the copies were made (see Section 8). To enable complete disk volume recovery there are also records indicating which datasets have been deleted, and when.

(iv) A pool of magnetic tapes for storing backups is available to the scheme. The software automatically selects tapes from this pool for reuse, removing this responsibility from operations staff. Each tape will contain part or all of a full backup or one or more partial backups.

(v) Operators have no responsibility for determining which backups to perform each night - that is, which disk volumes are to be fully dumped and which partially. This is also selected by the software on a cyclic basis.

(vi) It is extremely simple to restore a single dataset. It can be selected from any backup tape and copied directly to the target disk volume. Recovery of a full disk is automated but relatively time-consuming. It involves rebuilding the volume piecemeal, extracting the latest relevant version of each live dataset from the backup tapes and restoring it to disk.

(vii) All data transfers (i.e. backups and restores) are performed by a program called DUMPRSTR(ref.2), obtained from the North Carolina State University some time ago to reorganize 3330 disk volumes. The DRCS part of the backup scheme software has been designed for maximum independence from the program that performs the data transfer. The intention is to eventually replace DUMPRSTR by a better supported program with the same functional capabilities. Similar products are now offered by IBM(ref.3) and Innovation Data Processing(ref.4) and the Computing Centre intends to evaluate these in the near future. Support for 3350 drives has been added to DUMPRSTR by system programmers at DRCS.

5. THE MAIN FEATURES OF DUMPRSTR

DUMPRSTR has three basic components, providing disk to disk, disk to tape and tape to disk data transfer capabilities. In all three modes of operation an entire disk volume or individually selected datasets can be processed. At DRCS we use the program extensively, for reducing fragmentation and reclaiming waste disk space (disk to disk), for backups (disk to tape) and for data recovery (tape to disk). The latter two processes are the only ones of interest in this paper.

5.1 Dataset backups

DUMPRSTR can transfer all datasets on a disk volume to tape (a full backup) or selected datasets only (a partial backup). In both cases the program first writes three files of control information to the tape,including the names of datasets dumped, their disk characteristics

and space requirements and their position on tape. The individual datasets follow, each preceded by a file mark. Each track of a dataset that contains actual data is written as a single tape block. Unused tracks are omitted from the tape to eliminate unproductive effort.

The only defect in this method of operation is the use of file marks to separate datasets on tape. Besides wasting a considerable proportion of the tape, the file marks are written under program control, so that the Operating System has no opportunity to write header and trailer labels for standard-labelled tapes. Therefore, in the backup system, where a single job may write several partial dumps to the same tape, one after the other, non-labelled tapes must be used. An accurate count must be kept of the number of datasets dumped in each partial backup, to identify the file number of the start of the second and subsequent dumps. This is a fairly simple process and presents no real problem.

However there is a potential problem in the construction of the JCL for a job producing multiple partial backups on a single tape volume. Obviously the first dump will start at sequence number one. The LABEL parameter on the tape DD statement in the first step will be LABEL=(1,NL). Suppose the first partial backup contains 39 datasets. A total of 42 file marks will therefore be written. However the Operating System knows of only one. The LABEL parameter on the tape DD statement in the second step must therefore be LABEL=(2,NL), which is where the Operating System thinks the tape is positioned. This allows the data to be dumped directly to the tape, without the need for rewinding and repositioning. The danger lies in the fact that if the tape is accidentally or forcibly unloaded and reloaded at any time during the job then it will not be repositioned to the correct point and the data will be overwritten (because the file sequence numbers in the JCL LABEL parameters do not reflect the actual file counts). This sequence of events must therefore be strictly avoided. Should it begin to occur the job must be rerun.

5.2 Data recovery

DUMPRSTR can restore a complete disk volume from a full backup on tape, either to the same or a different volume serial number, provided it is of the same device type. Alternatively selected datasets can be restored from either a full or partial backup to any volume of the same type as that dumped. The first file sequence number of the backup containing the datasets must be specified in the JCL. For instance, suppose that the backup containing the selected datasets is the second on a tape, and that the first contains 39 datasets. Then the required file sequence number is 43 (to bypass the 42 files that comprise the first backup).

When a complete disk volume is restored any data already on the receiving volume is lost, as the original VTOC is overwritten. However, DUMPRSTR handles each selected dataset in a partial restore individually, in a manner similar to dataset creation. The restore may fail for any of the reasons a conventional dataset creation may fail - insufficient space on the volume, duplicate dataset name or no free entries in the VTOC. In addition DUMPRSTR will only restore unmovable datasets to the same absolute location on the receiving volume. If this area is already occupied then the restore will also fail. Indexed sequential (ISAM) and VSAM datasets are treated as unmovable by DUMPRSTR.

6. SMF DATA CAPTURE

The backup scheme relies on SMF data(ref.8) to determine when disk datasets are created, updated and deleted. This information dictates which datasets are to be included in partial backups and is recorded in the backup catalogue

for recovery purposes.

The SMF records used are -

(i) type 15 - creation and update of a non-VSAM dataset.

(ii) type 17 - deletion of a dataset (VSAM and non-VSAM).

(iii) type 18 - renaming of a non-VSAM dataset or a VSAM data or index component (assuming that the VSAM cluster was allocated with the UNIQUE attribute(ref.6), which is an installation standard). This is treated as a deletion of the old dataset and creation of the new.

(iv) type 63 - creation of VSAM data and index components.

(v) type 64 - update of a VSAM data and index component.

Because SMF data is not generated for started tasks dataset updates performed by them will not be detected. This limitation has not proved serious in the DRCS installation, since started tasks are not frequently used. Even when they are used they seldom access datasets on user disks and rarely, if ever, update them.

There is one other situation in which SMF records are not generated for dataset activity. This is when space is allocated to a new dataset which is never opened. Again this is not considered a problem, since the dataset is empty.

The backup procedure must have access to all SMF records of the five types mentioned that have been generated since the last run.

Some of these would normally be on tape, as the result of dumping SMF datasets SYS1.MANX and SYS1.MANY, and some still in one or both of these datasets. To avoid mounting the SMF tape during the backup run the SMF dump program copies the five record types to a disk dataset (SYS1.BACKUP.SMF) as well as to tape. The backup procedure can therefore obtain the information it requires from the three disk datasets mentioned. A separate procedure clears out SYS1.BACKUP.SMF when the backups have completed successfully, ready for the next day's data (see Appendix III.4).

7. SPECIAL CONSIDERATIONS FOR VSAM DATASETS

As mentioned in Section 6(v) SMF record type 64 is used to determine VSAM data and index component updates. The record in fact is written for any access, read or write. It contains a statistics section indicating the number of each type of access performed since the component was last opened. If none of the deletion, update or addition counts are positive the data has apparently been used for input only. While this assumption is sound for VSAM data components it is not reliable for index components. Only the addition count is maintained in the statistics section of the type 64 record for an index, probably because VSAM does not consider that it was truly opened. The user is really only processing the data component. Therefore index records may be updated without indication in the SMF record. For this reason the backup scheme must assume that all index components identified in type 64 records may have been updated. There is no harm in unnecessarily backing up those that have not changed, and the overhead is slight, since most index components are very small(usually only a single track).

VSAM datasets are inseparable from their catalogue entries. However it is not essential that the statistics portion of the catalogue entries be correct, as long as the extent descriptions are. A VSAM component can easily be restored if the copy has the same extents as those reflected by the catalogue entry of the current dataset by using the following procedure:

First scratch the dataspaces of the current component using the IEHPROGM

utility program(ref.7), but do not remove the catalogue entry. Then simply restore the copy. It will automatically occupy the same locations, since DUMPRSTR treats it as an unmovable dataset.

If the extent descriptions are different the dataset will be much more difficult to recover. In this case the procedure is:

First scratch the dataspaces and then restore the component. This may involve shifting datasets that occupy areas required for the copy, which may be a difficult task in itself. Then, when the catalogue containing the entries is not in use, generally outside the period of normal operations, take a selective backup of the catalogue and delete its dataspace. Restore the backup copy of the catalogue taken at the same time as the copy of the component. Reset the two VSAM timestamps in the format 4 DSCB of the catalogue volume to match the stamp in the volume record of the restored catalogue(ref.5). Now use the IDCAMS utility program(ref.6) to unload (REPRO) the VSAM dataset to tape. Next reset the VSAM timestamps to their original value, scratch the catalogue dataspace and restore the current version from the backup taken beforehand. Finally the VSAM dataset can be reloaded from tape.

Obviously this latter form of recovery should only be used in emergencies, when there is no other means available. Owners of VSAM datasets should always arrange periodic backups themselves.

8. THE BACKUP CATALOGUE

The backup scheme uses a VSAM key-sequenced cluster (SYS1.BACKUP.CATLG) to identify the current contents of the backup tapes. There is one record in the catalogue for each copy of each dataset currently on tape, plus records for deleted datasets. The records are each 80 bytes long and have the following format -

Offset	Length	Field
0	1	flag byte - see (i)
1	44	dataset name - see (ii)
45	6	disk volume - see (iii)
51	5	backup date - see (iv)
56	6	first tape volume - see (v)
62	2	file sequence no. - see (v)
64	6	second tape volume - see (v)
70	6	third tape volume - see (v)
76	4	backup time - see (iv)

Notes

(i) The flag byte is currently used for two purposes. As mentioned, there are two different record types in the catalogue - those that indicate the location of a dataset copy and those that indicate that a dataset has been deleted. The latter are used only during complete volume recovery, to ensure that deleted datasets are not restored. If the

second bit of the flag byte is on then this is one of those records.

The other use of the flag byte is to identify which dataset copies are unmovable. Those that are will have the first bit of the byte on. Again this indicator is used only during complete volume recovery(see Section 11.3).

The remaining six bits are currently not used and are always set to zero. The three possible values of the flag byte are therefore

'01000000'B - indicates a deletion record
'00000000'B - indicates a copy of a movable dataset
'10000000'B - indicates a copy of an unmovable dataset

(ii) The name field usually contains a dataset name, but it may also contain one of two special character strings - a single blank character followed by either 'PART' or 'FULL'. These records identify which partial and which full volume backups are currently available on tape, and always appear at the beginning of the catalogue for easy scanning, because of their low key value (see (vi) below). In all other respects these special records are the same as those of the datasets that form the backup.

(iii) The disk volume field identifies which volume the dataset came from (for a dataset copy record), which volume was fully or partially backed up (for one of the special records mentioned in (ii) above), or from which volume a dataset was deleted (for a deletion record).

(iv) The backup date is in Julian form, with numeric display attributes. The time indicates the time of day, as hhmm, that the backup procedure was run, and is also in numeric display format. It is included in each record to distinguish between two copies of a particular dataset that may have been made at different times on the same day.

(v) The three tape volume fields and file sequence number identify the location of either a full or partial disk volume backup, and hence the datasets it contains. Full backups may occupy one, two or three tape volumes, exclusively. However, partial backups should occupy a single tape volume, which may contain other partial backups (see Appendix I.3 (b)). Unused tape volume fields will contain blanks.

The sequence number (2 bytes, binary) identifies the position of the first file of the backup on the first or only tape volume. This will always be 1 for a full backup, but will be greater than 1 for the second and subsequent partial backups on a single volume (see Section 5.2).

(vi) The key of each 80-byte record comprises the dataset name, disk volume, backup date and first tape volume fields, a total of 61 bytes. This is the minimum required to uniquely identify any combination of dataset backups that may occur. For instance, the same dataset name may occur on more than one disk volume, or it may be backed up more than once on the same day.

Obviously the catalogue is essential to the backup scheme. In case of accidental damage it is copied to another dataset (SYS1.BACKUP.COPYCAT) on another disk each night. All updates to it during the past fortnight are also kept on backup tapes. Their locations can easily be determined from reports produced by the backup runs.

The loss of both SYS1.BACKUP.CATLG and SYS1.BACKUP.COPYCAT however would require more difficult recovery action. It would involve restoring a copy of the catalogue up to a week old and reapplying as many as five sets of updates

to it.

9. SELECTING BACKUP TAPES

The method by which tapes are selected for reuse has been designed to keep the size of the tape pool required to a minimum. The program that performs the selection is BACKUPS and the main points of the algorithm are repeated in the Appendices where appropriate.

When program BACKUPS has determined how many tapes it will require to perform the backup requests (see Appendix I.3 (b)) the next task is to decide which serial numbers to use. The complete set of eligible numbers is available to the program through an input dataset.

The first tapes selected are those that contain full backups that have expired. There must always be at least one full backup of each disk volume as old as the minimum retention period(currently set at six weeks). If there are more then the tapes containing the oldest ones are selected for reuse, provided that their serial numbers still appear in the tape pool. If they have been removed from the pool the tapes will not be reused.

If this first stage of the algorithm fails to yield the required number of volumes any tapes that are currently not in use are selected next. These might be volumes that have recently been added to the pool, for example. If still more tapes are required those that contain the oldest available partial backups are selected, again provided that their serial numbers still appear in the tape pool.

Note that this technique provides protection only for full backups. The onus is on the data manager to ensure that the pool always contains sufficient tapes to accomodate the desired retention period for partial backups, allowing a few extras in case of retries. If not, partial backups will be overwritten sooner than expected. It is desirable to periodically monitor the numbers of full and partial backups currently supported by the tape pool by listing the special records at the beginning of the backup catalogue (see Section 8(ii)).

10. THE BACKUP CYCLE

The other factor that influences the number of tapes required in the pool is the frequency of taking full backups. It is obviously essential to always have available all partial backups of a disk taken since the last full backup, in case volume recovery is necessary.

The backup scheme allows the data manager to define a series of backup specifications that will be processed in a cyclic manner, one each night (see Appendix I.3 (b)). The period of the cycle is decided by the manager, as are the backups to be performed at each stage. For example, the cycle in use at DRCS has a period of ten, which represents one working fortnight. There are currently six user 3350 disk volumes handled by the scheme. In six of the ten stages one of the disks is fully backed up and the others partially, while in the remaining four stages only partial backups are done. This means that each volume is backed up fully only once a fortnight, so that at least the last nine partial backups must always be available. The tape pool currently has 75 volumes, which is sufficient to retain four full backups of each disk, each occupying two tapes, as well as the last 13 partial backups, assuming there have been no retries. There are usually two or three partial backups per tape.

The automatic backup cycle may be circumvented and a specific set of backups executed instead. However, if this option is exercised the tape requirements should be checked, for the effect is identical to increasing the period of the cycle.

11. OPERATIONAL PROCEDURES

11.1 The Backup Process

The catalogued procedures used to run the backup scheme have been designed to be resistant to operator error. In addition, there are several supporting procedures to recover from a variety of error conditions. Appendix III contains a detailed description of each procedure.

(i) To initiate the backup process, first execute procedure BACKUPS. Normally no parameters are required. This job will automatically determine which disk volumes to backup fully and which to backup partially. It will also decide which backup tapes to use, relieving the operators of two book-keeping tasks. Immediately before completion, the job will submit one or more other jobs to the internal reader. There will typically be 2 or 3 of them, and the job names will be OPSBCK01, OPSBCK02 etc. These jobs perform the actual backups and will ask for the required tapes to be mounted.

(ii) Take care that the correct tapes are loaded. They are unlabelled tapes (see Section 5.1), so there is no Operating System check.

(iii) Under no circumstances should a tape be unloaded and then remounted or repositioned during the running of the OPSBCKnn jobs (see Section 5.1). If this accidentally happens requeue the particular job and rerun it.

(iv) As each full or partial backup ends one of the following messages will be displayed on the operator's console -

{ FULL / PARTIAL } BACKUP OF XXXXXX SUCCESSFUL

or

{ FULL / PARTIAL } BACKUP OF XXXXXX FAILED - REPLY U TO CONTINUE

If any backup fails note the disk serial number and whether it was a full or partial backup. In case the cause of failure may have been temporary the backup should be retried once, using catalogued procedure BACKOVER (see Appendix III.3) with parameter MEMBER of the form PXXXXXX (partial backup retry) or FXXXXXX (full backup retry), where XXXXXX is the disk volume serial number. The procedure will generate and submit another OPSBCK01 job to perform the retry. If any retry fails then the error is probably more serious. In such a case do not continue with step (v).

(v) When all backups (and retries, if any) have successfully completed run the procedure BACKUPOK. This will update system datasets in readiness for the following night's backups. Note that the backup process is restartable from step (i) at any stage before BACKUPOK is run. However, after running BACKUPOK selected backup retries (see (iv)) can still be performed but procedure BACKUPS cannot be restarted.

If by accident BACKUPOK is run before BACKUPS (i.e. step (v) is

run before step (i)) then recovery is accomplished by first initiating an SMF dump and, when it finishes, executing procedure RECOVER instead of BACKUPS. The backup process can then be resumed from step (ii).

(vi) The final step is to check the output of each partial backup. If there was other activity on the machine while the backups were running then datasets may have been deleted between the time they were selected for inclusion in the backup and the running of the actual DUMPRSTR job to transfer them to tape. There will be a DUMPRSTR error message whenever this occurs and program UPBACK (see Appendix II) should be used to decrease the file sequence number fields in the backup catalogue records pertaining to any subsequent partial backups on the same tape volume. Program BACKDEL should also be used to remove the catalogue records for the deleted datasets.

11.2 Dataset Restoration

The catalogued procedure RESTDSN can be used to restore a single dataset. It allows the following parameters for selection.

DSN(mandatory) identifies the dataset.

VOLUME(optional) identifies the disk volume on which the dataset resides or resided. This parameter should only be required if two copies of the dataset existed at backup time on the day in question.

TOVOL(optional) identifies the disk volume to which the dataset will be restored. If not specified, the dataset will be restored to the volume the selected copy came from. If TOVOL is specified it must identify a volume of the same device type as the one the copy came from.

DATE(optional) identifies the date (in Julian form) of the required backup copy. The copy taken on, or closest to and preceding this date that satisfies the other selection criteria will be chosen. DATE defaults to the run date, so that the most recent copy will be selected if this parameter is not supplied.

TAPE(optional) identifies the backup tape volume containing the copy required. This parameter may be necessary if the dataset was backed up more than once on the same day. If TAPE or COPY are not specified under these circumstances, the latest backup on that day is chosen. Note that full backups may occupy more than one tape volume. In this case TAPE should indicate the first volume.

COPY(optional) identifies the number of the required copy(see (ii) below).

The only step in the procedure is named RESTORE, and it executes the program RESTDSN (see Appendix II).

To restore a dataset the following steps are necessary -

(i) Delete and uncatalogue (or rename) the dataset to be restored,

if it still exists on disk. If this is not done the restore will fail.

(ii) If unsure what copies are available use the TSO command procedure COPIES to list the details of each. Any information displayed, including the copy number, may be used as a selection parameter for RESTDSN. See Appendix IV for further details.

(iii) Execute the RESTDSN procedure with DSN plus any other parameters necessary to identify the backup copy. The procedure will select the backup copy that satisfies the criteria. It will then submit a batch job named OPSREST to perform the actual data transfer. This job will be automatically held in the input queue and will only execute after operator intervention (see step (v)). The procedure will also produce a listing similar to the output from the COPIES command procedure and will indicate which copy was selected. If no copy satisfied the selection criteria then the OPSREST job will of course not be submitted.

(iv) If the copy selected is not the one required then ask the operator to cancel the OPSREST job. Next perform step (iii) again, this time supplying parameters that will select the required copy.

(v) If the correct copy was selected give the operators authority to release the OPSREST job. When it completes execution the dataset will be available for use.

Note that there are special considerations for recovering VSAM dataspaces. Section 7 describes the process in more detail.
The following two examples show typical uses of procedure RESTDSN.

(a) To restore the latest copy of dataset ABC.A.DATA

```
// EXEC RESTDSN,DSN='ABC.A.DATA'
```

(b) To restore the version of ABC.A.DATA as it existed at backup time on 5/8/77 (day 77217) to volume SA0004

```
// EXEC RESTDSN,DSN='ABC.A.DATA',DATE=77217,TOVOL=SA0004
```

11.3 Volume Recovery

The catalogued procedure used in volume recovery is RESTVOL, which executes the program of the same name (see Appendix II). The available parameters are as follows.

VOLUME(mandatory) identifies the disk volume to be restored.

TOVOL(optional) identifies the serial number of the target disk volume. The device type of this volume must be the same as that of the original. If not specified TOVOL defaults to VOLUME.

DATE(optional) identifies the date (in Julian form) to which the volume contents must be restored. No dataset updates occurring after this date will be incorporated. DATE defaults to the run date, so that the latest available version of the volume's contents will result if this parameter is not supplied.

TIME(optional) identifies the time of day (as hhmm), on the indicated date, to which the volume contents must be restored. This parameter should only be required if more than one backup of the volume was taken on the date in question, and not all of them are to be included in the recovery operation.

The steps required to initiate the recovery action are as follows -

(i) Reinitialize a disk volume with the serial number required. The VTOC must be in the same place as it was on the damaged volume.

(ii) Execute procedure RESTVOL with whatever parameters are necessary. This procedure first determines the latest full backup of the volume taken on or before the required date and time. It then uses the catalogue records for this backup, plus those for all partial backups taken in the intervening period, to determine the location of the latest relevant copy of each dataset on the volume. Catalogue deletion records identify datasets that have been deleted and therefore no longer exist on the volume.

Finally the procedure generates and submits a batch job containing a series of DUMPRSTR job steps to extract the live datasets and rebuild the volume. There will be up to two steps for each tape volume containing datasets involved in the recovery. On the first pass of the tapes only the selected unmovable datasets will be restored (including VSAM). This ensures that the areas they require on disk are not used by any other datasets. On the second pass the remaining datasets are extracted, completing the rebuilding process.

(iii) When the job has ended the volume can be relabelled, if required. The two VSAM time-stamps in its format 4 DSCB(ref.5) should also be updated if the volume contains VSAM dataspaces, to coincide with that in the volume's Operating System catalogue record.

12. SUMMARY

The operational and functional details of a selective dataset backup scheme have been presented. The increasing overhead associated with the direct backup techniques formerly used and the need for easier dataset recovery together emphasized the need for a new approach to data backup.

Although the new scheme continues to operate within the bounds of the design objectives there are two areas where improvement may be possible. Some direct access space on system volumes and elapsed time during the backup run could be saved by changing the organization of the backup catalogue from VSAM key-sequenced to physical sequential. The only direct access to the catalogue occurs during dataset and volume recovery and exceptional catalogue maintenance. These tasks would therefore suffer with an increase in execution time requirements. Their frequency of use will need to be monitored before making a decision to change the organization. However there are already signs that the use of the dataset recovery feature of the backup scheme will reach a level that warrants keeping the VSAM organization.

The second area that warrants further investigation has already been mentioned in Section 5. It would be desirable to replace the program DUMPRSTR by another with better performance and better vendor support, provided one can be found that meets all other functional requirements. The anticipated benefits of such a change are reductions of execution time and of the risk

that tapes will be overwritten. The new program should not write file marks between datasets on tape; this would permit the efficient use of standard-labelled volumes.

13. ACKNOWLEDGEMENT

The author wishes to thank Miss M.A. Kellaway of Computing Services Group for providing the majority of the programming effort for the project.

REFERENCES

No.	Author	Title
1	J.C. Gwatking	'A Data Migration Scheme for the 370/168 Computer at W.R.E.' WRE Technical Report 1870(A)
2	North Carolina State University	'DUMPRSTR, Dumping and Restoring Direct Access Volumes', Document No. LS-290-1
3	IBM Corporation	'OS/MVT, OS/VS DASDR Users Guide', SH20-9111
4	Innovation Data Processing	'Fast Dump Restore and Data Set Function', FDR/DSF V4 User Documentation
5	IBM Corporation	'OS/VS2 System Programming Library: Debugging Handbook', GC28-0709-0
6	IBM Corporation	'OS/VS2 Access Method Services', GC26-3841
7	IBM Corporation	'OS/VS Utilities', GC35-0005
8	IBM Corporation	'OS/VS2 System Programming Library: System Management Facilities (SMF)' GC28-0754-0

GLOSSARY

Several terms and abbreviations used in this report are unique to IBM computer systems. Although most are satisfactorily explained in the appropriate IBM publication brief descriptions are included here for quick reference.

Catalogued Procedure.
: This is a set of job control statements stored in an Operating System library and identified uniquely by a name of from 1 to 8 characters. The contents of this set can be included in a batch job by reference to this name. Parameters may be available to tailor the statements to individual needs.

Command Procedure.
: This is a set of TSO terminal commands stored in an Operating System or user library and identified by a name of from 1 to 8 characters. Execution of the procedure name causes execution of all commands in it. Parameters may be available to tailor commands to individual needs and control logic paths through the procedure.

DD(data definition) statement.
: A job control statement that associates a dataset with a file name defined in an application program.

DSCB(Dataset Control Block).
: This is one of a number of 144-byte records that reside in the VTOC of a direct access (disk) volume. There are several different types or formats, each with a different purpose. For example, a format 1 DSCB identifies the name, internal format and location of a dataset on the volume. There is one format 1 DSCB for each dataset. However there is only one format 4 DSCB in a VTOC, containing volume dependent information.

JCL(Job Control Language).
: This is a command language by which a batch job conveys its requirements and distinguishing characteristics to the Operating System.

JES(Job Entry Subsystem).
: This is a component of the Operating System that controls the interpretation of the JCL and processing of jobs.

LABEL parameter.
: This is a parameter of a DD statement that identifies which label options are in effect for a dataset. The options include whether labels are present or not (tape datasets only), the position on tape of a dataset and an expiry date or retention period (any dataset).

SMF(System Management Facilities).
: This is a component of the IBM Operating System that gathers and records details of system events. The data can be used for a variety of purposes, including accounting, performance monitoring and activity reporting.

TSO(Time Sharing Option).
: This is a component of the Operating System that provides a wide range of processing functions to a user at a terminal.

VSAM(Virtual Storage Access Method).
: A type of dataset organization that can provide direct as well as

sequential access capabilities, depending on application program requirements.

VTOC(Volume Table Of Contents).
This is a special file on a direct access (disk) volume that contains information about the volume and the datasets on it. Its records are called dataset control blocks (DSCB).

APPENDIX I

PERMANENT DATASETS

This appendix describes the purpose and contents of each permanent dataset used by the backup scheme software. Many of these datasets provide mechanisms to control or adjust different aspects of the system.

I.1 SYS1.BACKUP.CATLG

This dataset is the current version of the backup catalogue. It is a VSAM key-sequenced cluster and the format of each record is fully described in Section 8. The dataset is the focal point of the system and is used by most programs. It is updated by program BACKUP in procedures BACKUPS, BACKOVER and RECOVER (at the time it is actually named SYS1.BACKUP.CATEMP). The maintenance programs UPBACK and BACKDEL may also update it. However these should be used only when necessary and only by the data managers who understand the effects of the update.

I.2 SYS1.BACKUP.COPYCAT

This is the previous version of the backup catalogue. It is updated at the beginning of procedures BACKUPS and RECOVER by deletion and re-creation from SYS1.BACKUP.CATLG.

I.3 SYS1.BACKUP.DATA

After the catalogue, this is the next most important dataset in the system. It is in fact a partitioned dataset with fixed-blocked, 80-byte records. The contents and function of each member are described below, in alphabetic sequence.

(a) BACKUP

This member contains the skeleton JCL used by program BACKUPS to build the DUMPRSTR jobstream to perform the backups.

(b) BACKUP01, BACKUP02 etc.

These members contain backup specification statements. One member is used each night to define which backup operations to perform. The next night the next member in the sequence will be used and so on until the last is reached. The cycle will then begin with BACKUP01 again. The number of members therefore dictates the period of the backup cycle, and can easily be altered. The only consideration when increasing the period is whether there will still be enough tapes in the pool to meet retention objectives (see member TAPES below).

The selection of which member to use is made by program BACKOPT, which retrieves and passes the information in it to program BACKUP. The catalogued procedures involved are BACKUPS, BACKOVER and RECOVER.

The backup specification statements themselves can take one of two forms. For a full backup the word 'FULL' must appear in bytes 1 to 4, followed by the disk volume serial number in bytes 6 to 11. For a partial backup request from one to ten volume serial numbers can be specified, beginning in byte 1 and separated by commas. Each serial number appearing on a single request card will be partially backed up to the same tape volume, if possible (see below).

For example, consider the following specification requests -

```
FULL SA0002
SA0003,SA0004,SA0006
SA0005
```

These state that volume SA0002 is to be fully backed up. In addition volumes SA0003, SA0004 and SA0006 are all to be partially backed up to successive sequence numbers on the same tape. Finally volume SA0005 is to be partially backed up to a tape volume of its own. If these are all 3330 volumes then three tape volumes will be needed to satisfy the request. However full backups of 3350 volumes may require two tapes (see program BACKUPS). Note also that partial backups of 3330's and 3350's may be intermixed on a single tape.

Note that the software currently contains no provision for allowing partial backup jobs to use continuation spools, so the data manager must ensure that each partial backup specification statement will require the use of only one tape. If a partial backup job does request a continuation spool the operator will cancel the job and use the procedure BACKOVER to repeat the uncompleted backup and initiate any others the job still had left to process.

Although this situation has not yet arisen it is almost certain to in the future, as the quantity of on-line data grows. If it becomes a problem the software will be altered to allow for multiple tape volumes in a partial backup job. The reason for not including the function from the outset is that the best implementation would involve changes to DUMPRSTR, which may not remain part of the software for much longer (see Section 12).

(c) COPYBACK

This is a set of IDCAMS control statements to delete SYS1.BACKUP.COPYCAT, re-create it from SYS1.BACKUP.CATLG and create an empty SYS1.BACKUP.CATEMP on the same volume as the latter. This member is used in procedures RECOVER and BACKUPS.

(d) EXCEPT

This member defines the partial backup exception input used by program BACKUP in catalogued procedures BACKUPS, BACKOVER and RECOVER. The control statements identify datasets that should never be backed up in a partial backup and those always to be backed up. They have the following format -

Offset	Length	Field
0	1	exception indicator (A for always, N for never)
1	1	unused (blank)
2	6	disk volume (may be blank, indicating the dataset may be on any volume)
8	1	unused (blank)
9	44	dataset name
53	27	unused (blank)

For example, consider the following statements.

```
N SA0003  SYS1.DUMMY
A SA0004  SYS1.VIP
A         ABC.A.DATA
```

These indicate that dataset SYS1.DUMMY, on volume SA0003, should never be included in a partial backup. In addition, if dataset SYS1.VIP is on volume SA0004 it should always be backed up, even if there were no update records for it. Finally, dataset ABC.A.DATA should always be backed up, provided it can be found on any of the volumes participating in this backup run.

The primary use of this member is to indicate that VSAM user catalogues should always be backed up, since they do not generate SMF update records.

(e) FXXXXXX

There is one of these members for each disk volume (XXXXXX) participating in the backup scheme. They each contain a single full backup request card for the indicated volume (see (b) above for the format). For example, member FSA0004 would contain the record

```
FULL SA0004
```

The members are used by procedure BACKOVER to retry a failing full backup for the volume.

(f) PXXXXXX

There is also one of these members for each disk volume (XXXXXX) participating in the backup scheme. They each contain a single partial backup request card for the indicated volume (see (b) above), and are used for partial backup retries by procedure BACKOVER.

(g) RENAME

This member contains IDCAMS control statements to delete SYS1.BACKUP.CATLG and rename the newly created SYS1.BACKUP.CATEMP and its components so that it becomes the current backup catalogue. The member is used at the end of procedures BACKUPS, BACKOVER and RECOVER, after the necessary catalogue maintenance has been performed on SYS1.BACKUP.CATEMP.

(h) RESTDSN

Member RESTDSN contains the skeleton DUMPRSTR jobstream used by program RESTDSN to recover a dataset.

(i) RESTVOL

This member contains the skeleton DUMPRSTR jobstream that program RESTVOL modifies to restore an entire disk volume.

(j) TAPES

A list of the serial numbers of tapes available to the backup scheme is stored in this member, one per 80-byte record. This

information is used by program BACKUP in procedures BACKUPS, BACKOVER and RECOVER to help select which tapes to use this run. There must always be sufficient entries to meet the retention objectives for both full and partial backups, plus a few extra in case retries are required during the usage cycle. Note that a retry (procedure BACKOVER) does not reuse the same tape as the original, failing attempt at backup, but instead selects the next available volume.

To add or remove tapes from the pool simply add or remove their entries from the list. A tape serial number might be removed, for instance, if it contains a backup that should be kept indefinitely.

(k) VERIFY

This member contains IDCAMS control statements to verify that SYS1.BACKUP.CATLG has been properly closed and to delete and reallocate space for SYS1.BACKUP.CATEMP. The member is used by procedure BACKOVER in preparation to updating the catalogue.

(l) XMPTJOB

The names of jobs whose dataset accesses are to be disregarded are contained in this member, one per 80-byte record. The information is used by program DSUPDTE in procedures BACKUPS and RECOVER.

Typical entries could include the names of system jobs that compress partitioned datasets. Although SMF update records are generated the data is still logically the same.

I.4 SYS1.BACKUP.NEXTMEM

This dataset has a single 80-byte record which contains the name of the next BACKUPnn member of SYS1.BACKUP.DATA to be used for backup specifications by program BACKOPT, in procedures BACKUPS, BACKOVER and RECOVER. If the member does not exist the program assumes that the end of the cycle has been reached and reverts to member BACKUP01. Altering the contents of this dataset therefore interferes with the normal sequence of the backup cycle, which may be desirable in some circumstances.

I.5 SYS1.BACKUP.NEWMEM

If program BACKOPT (procedures BACKUPS, BACKOVER and RECOVER) uses the member name in dataset SYS1.BACKUP.NEXTMEM to select backup specifications it increments the name and stores it in SYS1.BACKUP.NEWMEM. Only after all backups have been successfully completed does catalogued procedure BACKUPOK copy the contents of this dataset into SYS1.BACKUP.NEXTMEM ready for the next backup run.

I.6 SYS1.BACKUP.LASTSMF

This dataset contains the timestamp of the last SMF record processed by the previous execution of the backup scheme. Only records generated after this time will be used by program DSUPDTE (procedures BACKUPS and RECOVER) to determine which datasets have been updated. If SYS1.BACKUP.LASTSMF is empty, then all records input to DSUPDTE through file SMFIN will be used.

The format of the single 80-byte record is -

Offset	Length	Field
0	4	time of day, in hundredths of a second, of the last SMF record (binary)
4	4	date of last SMF record (Julian format, packed decimal)
8	72	unused (blanks)

I.7 SYS1.BACKUP.NEWSMF

Program DSUPDTE (procedures BACKUPS and RECOVER) writes the timestamp of the last SMF record it processed to this file (see I.6 above for the format). At the successful completion of the backup run procedure BACKUPOK copies this dataset to SYS1.BACKUP.LASTSMF, ready for the next backup run.

I.8 SYS1.BACKUP.SMF

This disk dataset forms part of the SMF input to program DSUPDTE in procedure BACKUPS (together with SYS1.MANX and SYS1.MANY). It is formed by the SMF dataset dump procedure and contains only the record types of interest to the backup scheme (see Section 6). Because it is on disk there is no need to mount SMF dump tapes to extract the required data, saving both elapsed and CPU time.

When all aspects of the backup run have been successfully concluded procedure BACKUPOK will drain SYS1.BACKUP.SMF. It will be reloaded by any subsequent SMF dump job for use by the next backup run.

If by chance the dataset is destroyed or drained before procedure BACKUPS is run then procedure RECOVER should be used instead.

I.9 SYS1.BACKUP.UPDATES

Program DSUPDTE (procedures BACKUPS, RECOVER) creates this dataset for later use by program BACKUP (procedures BACKUPS, BACKOVER, RECOVER). It contains a summary of the last update action performed on each dataset encountered in the input SMF data, and is sorted by dataset name within volume serial number (both ascending).

The format of each 60-byte record is -

Offset	Length	Field
0	1	last action performed (S for scratch or delete, and U for update or create)
1	6	disk volume serial number
7	44	dataset name
51	4	date of the last update activity (Julian form and packed decimal)
55	4	time of day of the last update activity (a binary number representing the time in hundredths of a second)

59	1	unused (blank)

I.10 SYS1.BACKUP.CATADD

Program BACKUP (procedures BACKUPS, BACKOVER and RECOVER) writes records to be added to the backup catalogue to this dataset. At the conclusion the records are sorted in preparation to updating the catalogue. The record format is identical to that of the catalogue records.

I.11 SYS1.BACKUP.COPY

This dataset is generated by program BACKUP in procedures BACKUPS and RECOVER. It contains a copy of the DUMPRSTR jobstream created to perform the requested backup operations.

APPENDIX II

PROGRAM MODULES

In this appendix the programs that constitute the backup scheme software are described in detail. They are separated into five groups, according to the function they perform - the backup operation, dataset recovery, disk volume recovery, catalogue maintenance and miscellaneous tasks. All modules are coded in PL/I.

II.1 The Backup Operation

(i) BACKOPT

This program determines which backup operations are to be performed this run and passes the specifications to program BACKUP (see below). As mentioned in Appendix I, the partitioned dataset SYS1.BACKUP.DATA contains several members with names of the form BACKUPnn. Each describes a different set of backup specifications. The first member name must be BACKUP01, the second BACKUP02 etc. The number of these members determines the period of the backup cycle.

The function of BACKOPT is simply to select the member to be used this run. Under normal circumstances the program receives the member name through file NEXTMEM. It then copies information from this member to file BACKUPS, increments the member name and writes it to file NEWMEM (e.g. BACKUP03 becomes BACKUP04). However, if the member received from file NEXTMEM is not found in the dataset the program assumes that it has reached the end of the backup cycle and uses member name BACKUP01 instead (i.e. it reverts to the beginning of the cycle).

This process of automatic member selection can be circumvented by specifying a member name through the PARM field. This member, rather than that in file NEXTMEM, will then be used. Under these circumstances the member name can be of any form and file NEWMEM will not be updated with an incremented member name.

Input formats -

(a) PARM field
An 8-byte member name may optionally be specified in this field.

(b) File NEXTMEM - dataset SYS1.BACKUP.NEXTMEM
The member to be used in the absence of any name in the PARM field.

(c) File EXCEPT - dataset SYS1.BACKUP.DATA
The partitioned dataset containing the selected member.

Output formats -

(a) File NEWMEM - dataset SYS1.BACKUP.NEWMEM
The incremented member name when file NEXTMEM is used to determine the backup specifications.

(b) File BACKUPS
This file contains the backup specifications selected by the program. The record type is fixed length, 80 bytes.

(c) File SYSPRINT
Confirmation of the member name used, its contents and error messages are written to this file.

(ii) DSUPDTE

DSUPDTE processes SMF data to identify which datasets have been

updated, created or deleted since the previous execution of the program. Accesses by certain exempt jobnames are ignored.

As the program scans the SMF update records (types 15,17,18,63 and 64) it determines what was the last update operation (write or delete) performed on each dataset/volume combination (i.e. allowance is made for datasets of the same name residing on different disk volumes). This information is written to file SMFOUT for later processing by program BACKUP. The records are sorted by dataset name within disk volume serial number, both in ascending sequence.

Finally the program writes the time-stamp of the last SMF record processed to file NEWSMF. This will be used next time DSUPDTE is executed to identify the range of SMF records to be used.

Input formats -

(a) File SMFIN
This file contains the SMF input data. It is generally a concatenation of SYS1.BACKUP.SMF, SYS1.MANX and SYS1.MANY (see Section 6). However any SMF dataset may be used, including those stored on tape.

(b) File XMPTJOB - dataset SYS1.BACKUP.DATA(XMPTJOB)
The names of the exempt jobs - i.e. those whose dataset accesses are disregarded.

(c) File LASTSMF - dataset SYS1.BACKUP.LASTSMF
This dataset contains the timestamp of the last SMF record processed by the previous execution of the program. Only records written after this time are processed during this run.

Output formats -

(a) File NEWSMF - dataset SYS1.BACKUP.NEWSMF
This contains the timestamp of the last SMF record processed.

(b) File SMFOUT - dataset SYS1.BACKUP.UPDATES
DSUPDTE writes a record, identifying the last change activity for each dataset, to this file.

(iii) BACKUP

Program BACKUP performs the major functions of the backup operation. It first reads the backup specifications passed by program BACKOPT and determines how many tape volumes will be needed. Next the actual serial numbers are selected. File TAPES contains a list of all the tape volume serial numbers available to the backup system. The program first selects tapes containing full backups that are no longer required (see Section 9). If this process does not yield the required number of tapes the remainder are obtained by selecting any tapes currently not in use and then by reusing those tapes that contain the oldest available partial backups.

Next each backup request is processed in turn. For a full backup the VTOC of the disk volume is read and one update record written to file CATADD for each dataset. This file will later be used to update the backup catalogue. A set of skeleton JCL is modified as necessary, inserting the correct disk and tape volume serial numbers, and submitted to the internal reader to perform the actual backup.

For a partial backup request the records pertaining to the volume are located in file UPDATES, which contains the update information generated and passed by program DSUPDTE (see (ii) above). Those records identifying datasets deleted from the volume are used to generate deletion records to file CATADD. Those identifying dataset updates, processed in conjunction with exception specifications, are used to generate update records. Once again the skeleton JCL is modified and submitted. DUMPRSTR control statements indicating the

datasets to be backed up are included.

When all backup requests have been processed the remaining task is to update the backup catalogue. This is a sequential process, with input file BACKCOP containing the current catalogue and output file BACKCAT initially an empty VSAM dataset. File CATADD contains the records to be inserted into the catalogue. In addition those records in file BACKCOP that contain the serial number of any of the tapes that were reused this run are deleted i.e. they are not copied to the output file. At the conclusion file BACKCAT contains the new backup catalogue.

Program BACKUP makes use of several routines from the DRCS data migration software(ref.1). These include DSNVTOC, DSCB1, ELAPSED, JULIAN, DEVSIZE and DEVFREE.

Input formats -

(a) PARM field : retention,volumes
Retention is the mandatory retention period for full backups of any disk volume (in days). There must always be at least one full backup as old as this and no younger full backup can be destroyed.
Volumes specifies the number of tape volumes to allow for full backups of 3350 disk volumes.

(b) File TAPES - dataset SYS1.BACKUP.DATA(TAPES)
The pool of tape volume serial numbers available for use by the backup system.

(c) File EXCEPT - dataset SYS1.BACKUP.DATA(EXCEPT)
The exception input for partial backups - i.e. those datasets always or never to be backed up.

(d) File BACKJCL - dataset SYS1.BACKUP.DATA(BACKUP)
The skeleton JCL used to construct the backup jobs.

(e) File UPDATES - dataset SYS1.BACKUP.UPDATES
The dataset update information generated by program DSUPDTE.

(f) File BACKCOP - dataset SYS1.BACKUP.CATLG
The current version of the backup catalogue.

(g) File SYSIN
This file contains the specifications of which backups are to be performed. The record format is fixed length, 80 bytes.

Output formats -

(a) File BACKOUT
This file is directed to the internal reader. It contains the entire jobstream generated to perform the backups.

(b) File BACKUPS - dataset SYS1.BACKUP.COPY
A copy of the entire jobstream generated to perform the backups.

(c) File CATADD - dataset SYS1.BACKUP.CATADD
The records added to the backup catalogue.

(d) File BACKCAT - dataset SYS1.BACKUP.CATEMP
The new version of the backup catalogue, with all additions and deletions applied.

II.2 Dataset Restoration

(i) RESTDSN

This is the principal program of the dataset restore process. It uses parameter input to select which copy of the dataset is required and then modifies a set of skeleton JCL statements and submits them to the internal reader to perform the actual data transfer (using program DUMPRSTR).

The parameter input that can be used for selection includes disk volume, backup date, copy number, and backup tape. The latest copy of a dataset that satisfies all specified conditions is the one that will be selected. The complete list of copies available is printed, with the selection indicated, in case another version is required.

Input formats -

(a) PARM field : dsn,[disk],[date],[tovol],[tape],[copy]
Dsn is the dataset required.
Disk is the disk volume it resided on at the time of backup.
Date is the Julian date of the required backup (this defaults to the latest version available if the parameter is omitted).
Tovol is the disk volume the dataset should be restored to. If omitted the default is the volume from which it came.
Tape is the tape serial number containing the required backup copy.
Copy is the copy number as indicated on the list produced by program BCOPIES in the TSO command procedure COPIES (see (ii) below and Appendix IV.1), or by program RESTDSN itself.

(b) File RESTJCL - dataset SYS1.BACKUP.DATA(RESTDSN)
The skeleton JCL used to construct the DUMPRSTR jobstream.

(c) File BACKCAT - dataset SYS1.BACKUP.CATLG
The current version of the backup catalogue.

Output formats -

(a) File RESTORE
This file is directed to the internal reader. It will contain the generated DUMPRSTR jobstream.

(b) File SYSPRINT
The list of available backup copies and an indication of the one selected.

(ii) BCOPIES

This program is invoked by the TSO command procedure COPIES to produce a list containing the relevant details of all available backup copies of a dataset or datasets. Any of the information presented can be used as selection criteria for RESTDSN, including the number of the copy.

Input formats -

(a) PARM field : dsn[,ALL]
Dsn is either a full dataset name or a name stem.
ALL is optional. If present it indicates that dsn is a name stem and details of all datasets that begin with these characters will be presented. If ALL is omitted, dsn is treated as a full dataset name.

(b) File BACKCAT - dataset SYS1.BACKUP.CATLG
The current version of the backup catalogue.

Output format -

(a) File SYSPRINT
The list of available copies is written to this file.

II.3 Volume Recovery

(i) RESTVOL

RESTVOL generates and submits a DUMPRSTR jobstream to automatically recover a volume by collecting the latest version of each dataset from one or more backup tapes.

Parameter input defines the volume to be recovered and the date and time it should correspond to. In the absence of the timestamp information the result will be the latest available version of the volume.

The first task for the program is to determine the latest full backup of the disk volume taken at or before the indicated (or defaulted) timestamp. If the selected backup was actually taken at the required time then recovery is achieved simply by restoring it in its entirety to the disk volume.

However, it is more likely that there will have been at least one partial backup of the volume taken between the full backup and the restore timestamp. If this is so RESTVOL extracts those records from the backup catalogue pertaining to the full backup and all relevant partial backups and sorts them into dataset name (ascending) and backup timestamp (descending) sequence. The first record for each dataset in the sorted output defines its final status. If it is a deletion record then the dataset did not exist on the volume at the required time. If it is an update record then it identifies the backup tape containing the latest relevant version of the dataset.

RESTVOL uses these update records to generate a DUMPRSTR job to restore all the datasets to the new volume. All unmovable datasets are restored from each backup tape first. Next there is a further step for each tape to restore the remainder of the datasets.

RESTVOL also makes use of module JULIAN from the data migration software(ref.1).

Input formats -

(a) PARM field : volume,[date],[tovol],[time]
Volume is the disk to be restored.
Date is the date, in Julian form, to which the volume contents must be restored. The default is the run date.
Tovol is the serial number of the receiving disk volume. The default is the volume serial number that is being recovered.
Time is a timestamp, specified as hhmm. It identifies the time of day on the indicated (or defaulted) date to which the contents of the volume must be restored. The default is midnight (2359).

(b) File RESTJCL - dataset SYS1.BACKUP.DATA(RESTVOL)
The skeleton JCL used to construct the DUMPRSTR job.

(c) File BACKCAT - dataset SYS1.BACKUP.CATLG
The current version of the backup catalogue.

(d) File TEMP
This is a temporary file only. It is used during the sort procedure. The fixed length records must be 80 bytes long.

Output formats -

(a) File RESTORE
This file is directed to the internal reader. It will contain the generated DUMPRSTR jobstream.

(b) File OUTPUT
This is also a temporary file used during sort. Again the record type is fixed length, 80 bytes.

(c) File SORTOUT
This is a third temporary sort file.

II.4 Catalogue Maintenance

(i) BACKDEL

This program deletes records from the backup catalogue. The full 61-byte record keys are read from file SYSIN. The program may be used for instance, to remove a backup from the system, by deleting

its identification record (see Section 8(ii)).

Input formats -
(a) File SYSIN
The keys are contained in the first 61 bytes of each 80-byte record.
(b) File BACKCAT - dataset SYS1.BACKUP.CATLG
The current version of the backup catalogue.

(ii) LSTBACK

LSTBACK formats and prints the entire contents of the backup catalogue. Because of the enormous amount of output it produces and the highly volatile contents of the catalogue the program is run only as required.

Input format -
(a) File BACKCAT - dataset SYS1.BACKUP.CATLG
The current version of the backup catalogue.

Output format -
(a) File SYSPRIN
The report is written to this file.

(iii) UPBACK

This program updates the file sequence number field of selected backup catalogue records. Parameter input defines a tape volume, base sequence number and update amount (positive or negative). All records in the catalogue with this tape serial number in the primary tape field and with an initial file sequence number greater than or equal to that specified as the base value have the latter field incremented by the update amount. If the sequence number becomes negative or zero the record is deleted from the catalogue, otherwise it is rewritten.

The main use of the program is to decrease the file sequence number field of partial backup records when a previous partial backup on the same tape failed to find all selected datasets. This may occasionally occur if the backups are performed when other jobs are running.

Input format -
(a) PARM field : tape,sequence,update
Tape is the backup tape serial number.
Sequence is the base file sequence number.
Update is the sequence number increment, positive or negative.

II.5 Miscellaneous

(i) BACKFUL

This program produces a list of the tape volume serial numbers containing the latest full backups of each disk volume found in the backup catalogue. Operations staff run the program weekly and remove the tapes to another building to ensure recoverability after a catastrophe at the central computer site.

Input format -
(a) File BACKCAT - dataset SYS1.BACKUP.CATLG
The current version of the backup catalogue.

Output format -

(a) Operator's console

The sorted list of tapes, plus instructions on what to do with them, is displayed on the master console, and also appears in the JES log section of the printed output.

APPENDIX III

THE BACKUP PROCEDURES

This appendix describes the function of each step in the catalogued procedures used to perform the backup task. The operating instructions were outlined in Section 11.1.

III.1 Procedure BACKUPS

This is the main procedure of the backup scheme. It is used to initiate the backups under normal conditions.

There are three optional parameters, which should be used only under specific instructions from the data manager. They are -

MEMBER(optional) identifies the name of a member in SYS1.BACKUP.DATA that contains the backup specifications to be used during this run. Using this parameter causes the normal backup cycle to be bypassed.

FULLVOL(optional) identifies the number of tape volumes to allow for full backups of 3350 disk volumes. The default is currently 2.

FULLDAY(optional) identifies the minimum age (in days) of the oldest full backup of each disk volume. The default is currently 42 days, or 6 weeks.

The name and function of each step in the procedure is detailed below. The names of programs described in Appendix II are indicated where appropriate.

(i) COPYBACK

This step executes the Access Method Services Program (IDCAMS) to delete the dataset SYS1.BACKUP.COPYCAT and re-create it from SYS1.BACKUP.CATLG. In addition an empty VSAM dataset called SYS1.BACKUP.CATEMP is formed.

(ii) OPTIONS - program BACKOPT

The backup specifications to be used during this run are selected in this step.

(iii) UPDATES - program DSUPDTE

This step processes SMF information from the datasets SYS1.BACKUP.SMF, SYS1.MANX and SYS1.MANY to determine which datasets have been updated since the last backup run.

(iv) BACKUP - program BACKUP

This is the major step. It selects the tape volumes to be reused and builds a DUMPRSTR jobstream to perform the actual data transfer. The jobstream is written to dataset SYS1.BACKUP.COPY, as well as to the internal reader. In addition the backup catalogue is updated, the temporary dataset SYS1.BACKUP.CATEMP containing the new version.

(v) RENAME

Provided all previous steps have executed successfully the

utility program IDCAMS deletes SYS1.BACKUP.CATLG and renames SYS1.BACKUP.CATEMP so that it becomes the current catalogue.

III.2 Procedure RECOVER

This procedure is predominantly the same as BACKUPS. It is used instead of BACKUPS when procedure BACKUPOK is run out of turn, thereby destroying the SMF data in SYS1.BACKUP.SMF. Under these circumstances an SMF dump must be initiated, and when it completes procedure RECOVER must be run.

The parameters and steps in RECOVER are the same as those in BACKUPS. The only difference is the source of SMF data in step UPDATES. RECOVER obtains all its data from the SMF dump tape.

III.3 Procedure BACKOVER

Procedure BACKOVER is an abridged version of procedure BACKUPS. It is used to retry a specific backup when the original attempt failed.

The parameters available are the same as in procedure BACKUPS, but with MEMBER a mandatory parameter. It identifies the member in SYS1.BACKUP.DATA containing the backup specification request for the operation that failed. The member name will be of the form PXXXXXX or FXXXXXX, where XXXXXX is the disk volume serial number and the prefix identifies the type of backup, partial (P) or full (F).

The steps in the procedure are as follows -

(i) COPYBACK

This IDCAMS step ensures that SYS1.BACKUP.CATLG, the current backup catalogue, is closed. Next it re-creates the empty dataset SYS1.BACKUP.CATEMP.

(ii) OPTIONS - program BACKOPT

The data contained in the selected member of SYS1.BACKUP.DATA is located and used as the backup specification for this run.

(iii) BACKUP - program BACKUP

This step selects the tape volumes to be used and builds a DUMPRSTR job to perform the backup. The job is named OPSBCK01 and is submitted directly to the internal reader. Unlike procedures BACKUP and RECOVER, the jobstream is not duplicated in SYS1.BACKUP.COPY. Note that different tape volumes will be chosen. The volumes used for the failing backup request will not be reused, as they may still contain useful data.

The backup catalogue is also updated to reflect the anticipated new location of the datasets involved in the retry. SYS1.BACKUP.CATEMP will contain the new version.

(iv) RENAME

Provided all previous steps have executed successfully IDCAMS deletes SYS1.BACKUP.CATLG and renames SYS1.BACKUP.CATEMP, so that it becomes the current catalogue.

III.4 Procedure BACKUPOK

This procedure should only be executed after all backups and retries (if any were necessary) have been completed successfully, or at the direction of the data manager. The procedure alters the contents of

certain of the system datasets in preparation for the next backup run. This makes restart of the current run (by executing procedures BACKUPS or RECOVER) difficult, and certainly not possible without the involvement of the data manager. However, until BACKUPOK is executed either of those two procedures can be restarted without requiring special action.

The steps in the procedure are -

(i) TALKBACK

This step is included only as a safeguard against running the procedure out of turn. It asks the operator to confirm his decision to start it.

(ii) SAVEMEM

Step SAVEMEM copies the contents of dataset SYS1.BACKUP.NEWMEM to SYS1.BACKUP.NEXTMEM. These datasets are used by program BACKOPT to identify the member name in SYS1.BACKUP.DATA to be used for the next set of backup specifications.

(iii) SAVESMF

This step copies the contents of dataset SYS1.BACKUP.NEWSMF to SYS1.BACKUP.LASTSMF. The two datasets contain SMF record timestamp information for program DSUPDTE.

(iv) EMPTYSMF

Step EMPTYSMF drains SYS1.BACKUP.SMF, the dataset containing pertinent SMF records for use by program DSUPDTE.

APPENDIX IV

TSO COMMAND PROCEDURES

Two command procedures are available under TSO to provide an easy means of restoring an individual dataset. They are intended for use by duty programmers responding to user requests for data recovery.

IV.1 Procedure COPIES

This procedure uses program BCOPIES to produce a numbered list of all available backup copies of a particular dataset or datasets. All information that could be used as selection criteria is included.

The only required input is the positional parameter DSN, identifying the dataset name in its fully qualified form. However, the keyword parameter ALL may be used to signify that DSN represents a name stem only, and that information about all datasets beginning with this stem is required. For example, to obtain information about all datasets beginning with ABC.A use the command -

```
COPIES   ABC.A   ALL
```

IV.2 Procedure RESTDSN

This command procedure builds and submits a batch job to execute the catalogued procedure of the same name to restore a dataset. The name of the job will be useridR where userid is the 3-character TSO user identification.

Input to the procedure includes the positional parameter DSN which is mandatory and identifies the dataset name in fully qualified form. All other parameters are of the keyword type and have the same name and meaning as the selection options for catalogued procedure RESTDSN. These are VOLUME, TOVOL, DATE, TAPE and COPY.

For example, to restore dataset ABC.A.DATA as it existed at backup time on 5/8/77(day 77217) to volume SA0004 use the command -

```
RESTDSN   ABC.A.DATA   DATE(77217)   TOVOL(SA0004)
```

DOCUMENT CONTROL DATA SHEET

Security classification of this page: UNCLASSIFIED

1 DOCUMENT NUMBERS

AR Number: AR-001-227

Report Number: ERL-0021-TR

Other Numbers:

2 SECURITY CLASSIFICATION

a. Complete Document: UNCLASSIFIED

b. Title in Isolation: UNCLASSIFIED

c. Summary in Isolation: UNCLASSIFIED

3 TITLE

A SELECTIVE DATASET BACKUP SCHEME FOR THE DRCS CENTRAL COMPUTER

4 PERSONAL AUTHOR(S):

J.C. Gwatking

5 DOCUMENT DATE:

July 1978

6 6.1 TOTAL NUMBER OF PAGES: 42

6.2 NUMBER OF REFERENCES: 8

7 7.1 CORPORATE AUTHOR(S):

Electronics Research Laboratory

7.2 DOCUMENT SERIES AND NUMBER

Electronics Research Laboratory 0021-TR

8 REFERENCE NUMBERS

a. Task: DST 20/14

b. Sponsoring Agency:

9 COST CODE:

228753

10 IMPRINT (Publishing organisation)

Defence Research Centre Salisbury

11 COMPUTER PROGRAM(S) (Title(s) and language(s))

12 RELEASE LIMITATIONS (of the document):

Approved for Public Release

12.0	OVERSEAS	NO		P.R.	1	A		B		C		D		E	

Security classification of this page: UNCLASSIFIED

Security classification of this page: UNCLASSIFIED

13 ANNOUNCEMENT LIMITATIONS (of the information on these pages):

No limitation

14 DESCRIPTORS:			15 COSATI CODES:
a. EJC Thesaurus Terms	Data backup Data recovery Computers Data processing Data retrieval Computer programs	Computer programming Computer systems programs	0902 0502
b. Non-Thesaurus Terms	Data backup		

16 LIBRARY LOCATION CODES (for libraries listed in the distribution):

SW SR SD AACA

17 SUMMARY OR ABSTRACT:
(if this is security classified, the announcement of this report will be similarly classified)

This report describes the approach taken by the DRCS Computing Centre to maintaining adequate backup facilities while the quantity of data stored on-line continues to increase. The aim is to provide the required level of service to users with minimum operational overhead. This has been accomplished by the development of a software system to control the backup process. The basis of this scheme is the ability to determine which datasets have been altered and to select only these for backup.